

REMARKS

In the Office Action, the Examiner rejected Claims 31-36, which were all of the then pending claims, under 35 U.S.C. 103 as being unpatentable over the prior art. In particular, Claims 31 and 33-36 were rejected as being unpatentable over U.S. Patent 6,078,744 (Wolczko, et al. in view of U.S. Patent 6,367,012 (Atkinson, et al, further in view of U.S. Patent 6,704,927 (Bak, et al.), and Claim 32 was rejected as being unpatentable over Wolczko, et al in view of Bak, et al.

Claims 31 and 34-36 were further rejected under 35 U.S.C. 101 and under 35 U.S.C. 112, first paragraph. The Examiner also objected to language in Claims 31 and 32 and rejected these claims under 35 U.S.C. 112, second paragraph, as being indefinite.

Claims 31-34, each of which is an independent claim, are being amended to better define the subject matters of these claims. Claims 31 and 32 are also being amended to address the rejection of these claims as being indefinite. The rejections of Claims 31 and 33-36 under 35 U.S.C. 101 and 112, first paragraph, are respectfully traversed. In addition, Applicants are herein adding new Claims 37 and 38, which are dependent from Claim 31, to describe preferred features of the invention.

For the reasons discussed below, Claims 31-36 fully comply with 35 U.S.C. 101 and 112 and patentably distinguish over the prior art. The Examiner is thus asked to reconsider and to withdraw the rejections of Claims 31-36 under 35 U.S.C. 103, the rejections of Claims 31 and 33-36 under 35 U.S.C. 101 and 112, first paragraph, and the rejection of Claims 31 and 32 under 35 U.S.C. 112, second paragraph. The Examiner is further asked to allow Claims 31-36 and new Claims 37 and 38.

In rejecting Claims 31 and 34-36 under 35 U.S.C. 101 and 112, first paragraph, the Examiner argued that the claims are inoperable and not enabled, because there is no provision for exception handling. For instance, Claim 31 describes verifying matches, but does not describe what happens if the matches are not verified. Also, Claim 34 describes a check to see if code has been changed, but the claim does not describe what happens if the code has been changed.

These rejections are respectfully traversed because it is not necessary for the claims to describe all possible or alternative actions that a user of the invention might want to take. The claims define the invention, they do not need to describe the invention in detail. In the case of Claim 31, the claim requires that the matches be verified. Specifically what happens if the matches are not verified is not part of the definition of the invention defined by Claim 31. Similarly, with respect to Claim 34, specifically what happens if the code changes is not part of the definition of the subject matter defined by the claim.

Those of ordinary skill in the art can readily practice the subject matters of each of Claims 31 and 33-36. They might, for various reasons, want to practice additional steps, but this does not negate the fact that they can practice what is defined by the claims. When claims 31 and 33-36 are viewed in this light, it is clear that the claims are operable and enabled by the specification. The Examiner is, accordingly, asked to reconsider and to withdraw the rejections of Claims 31 and 33-36 under 35 U.S.C. 101 and 112, first paragraph.

Claims 31 and 32 are being amended to address the rejection of these claims under 35 U.S.C. 112, second paragraph. To elaborate, in the Office Action, the Examiner noted that the terms "the generated code" in Claims 31 and 32 and "the default action" in Claim 32 did not have the appropriate antecedent basis, and the Examiner suggested changing "generated code" to "precompiled code" and changing "byte code" to "intermediate representation."

In response, Applicants' Attorneys have carefully reviewed Claims 31 and 32 and are making editorial changes to use terms more consistently. For Example, in these claims, "generated code" is being changed to "precompiled code," which is introduced in the preambles of these claims.

A careful review of Claims 31 and 32 shows that these claims are clear and definite, and in particular, shows that terms are used consistently in the claims and are appropriately introduced into the claims. Accordingly, these claims fully comply with the requirements of 35 U.S.C. 112, second paragraph, and the Examiner is asked to also reconsider and to withdraw the rejection of Claims 31 and 32 under 35 U.S.C. 112, second paragraph.

With respect to the rejections of Claims 31-36 under 35 U.S.C. 103, Applicants respectfully submit that these claims patentably distinguish over the prior art because the references of record do not disclose or suggest the feature of annotating programs with fine grain dependencies, and processing those dependencies by use of a dependence granularity adjustor, as described in independent Claims 31, 32, 33 and 34. In order to best appreciate this, it may be helpful to summarize briefly this invention and the prior art.

As explained in detail in the present application, this invention relates to procedures to compile programs or components of a program in a mixed static and dynamic environment. Static compilation involves the process of translating in an off-line manner and generating one or more binary codes to be executed at run-time. In contrast, dynamic compilation involves translating a program component to machine code at run-time, before executing that program component. With prior art systems, several difficulties are encountered when implementing dynamic and static compilation. In particular, there are problems with performance overhead, dynamic binding and dynamic class loading, testability and serviceability of compilation,

The present invention effectively addresses these problems by use of a two step process. In a first step, a compiler is used to perform one set of tasks; and then, in a second step, a virtual machine is used to perform additional tasks. In that first step, pre-compiled programs are saved, including determining where to place those programs, annotating the programs with dependence information, and processing the programs to produce a further annotated executable code with annotations to help adapt the code to a new executable environment. In addition, these programs are annotated with fine grain dependencies, and those dependencies are processed, using a dependence granularity adjuster, to replace some fine-grain dependencies by coarser-grain dependencies to produce a final list of dependence annotations.

Claim 31 is specifically directed to a method for using a virtual machine to execute securely statically compiled code. As described in this claim, a compiler performs a first set of integrity checks, and the virtual machine conducts a second set of integrity checks.

Claim 32 is directed to a procedure for linking precompiled code at run-time within a virtual machine. In this procedure, the compiler maintains certain symbolic entries, and the virtual machine uses these symbolic entries, before the code is executed, to generate direct references in the generated code.

Claim 33 defines a method for updating code at run time, when separately compiled code, which contains symbols, changes. With this method, the compiler generates certain code, and the virtual machine may be used to recompile that code under defined circumstances.

Claim 34 is drawn to a method for maintaining compliance with a language requiring dynamic compliance, while still enabling the use of statically generated code for some byte code that depends on byte code that may be separately compiled. With this method, the compiler

performs security features on byte code, and the virtual machine uses those security features to determine if the byte code has changed.

An important feature of the present invention, described in each of Claims 31-34, is the use of precompiled programs, and what is done with those programs. More specifically, each of Claims 31-34 describes the step of saving pre-compiled programs, including determining where to place those programs, annotating the programs with dependence information, and processing the programs to produce a further annotated executable code with annotations to help adapt the code to a new executable environment. Each of these claims also describe the feature that the step of annotating the programs with dependence information includes the steps of annotating the programs with fine-grain dependencies, and processing said fine-grain dependencies by a dependent granularity adjuster to replace some fine-grain dependencies by coarser-grain dependencies to produce a final list of dependent annotations.

Independent Claims 31-34 all differ from Wolczko, et al. in that Wolczko does not disclose the use of precompiled programs, as described in Claims 31-34. These claims describe the use of precompiled programs, not analysis information. The use of precompiled code allows for a better recompilation into native code. As a result, unlike Wolczko, the present invention can dispense with having a compiler in the execution engine.

Atkinson, et al. was cited for its disclosure of a method and system to help ensure file integrity. This reference teaches incorporating a certification or signature in a file. This certification or signature may be confirmed at the recipient computer.

In order to address the deficiencies of Wolczko and Atkinson, et al. as references, the Examiner cites Bak, et al for its disclosure of annotating pre-compiled code with dependent and


dependence information. Bak, et al, however, does not disclose or suggest the use of fine-grain dependence information, as described in Claims 31-34.

This feature of the invention is important because it helps to combine effectively static and dynamic compilation. As a result of this, the invention achieves the reduced performance overhead of dynamic compilers while, at the same time, also achieving the aggressiveness that can be achieved with static compilers.

In light of the above-discussed differences between Claims 31-34 and the prior art, and because of the advantages associated with those differences, it cannot be said that any of these claims is obvious in view of that prior art. Accordingly, Claims 31-34 patentably distinguish over the prior art and are allowable. Claims 35 and 36, as well as new Claims 37 and 38, are dependent from Claim 31 and are allowable therewith. The Examiner is thus asked to reconsider and to withdraw the rejections of Claims 31-36 under 35 U.S.C. 103, and to allow Claims 31-38.

Every effort has been made to place this application in condition for allowance, a notice of which is requested. If the Examiner believes that a telephone conference with Applicants' Attorneys would be advantageous to the disposition of this case, the Examiner is asked to telephone the undersigned.

Respectfully Submitted,


John S. Sensny
Registration No. 28,757
Attorney for Applicants

Scully, Scott, Murphy & Presser
400 Garden City Plaza-Ste 300
Garden City, New York 11530
(516) 742-4343

JSS:ab